(72) Inventors: DULUK, Jerome, F., Jr.; 950 North California Avenue, Palo Alto, CA 94303 (US). HESSEL, Richard, E.; 3225 Flemington Court, Pleasanton, CA 94588 (US). ARNOLD, Vaughn, T.; 621 Canepa Drive, Scotts Valley, CA 95066 (US). BENKUAL, Jack; 11661 Timber Spring Court, Cupertino, CA 95014 (US). BRATT, Joseph, P.; 1045 Oaktree Drive, San Jose, CA 95129 (US). CUAN, George; 798 Lusterleaf Drive, Sunnyvale, CA 94086 (US). DODGEN, Steven, L.; 15735 Forest Hill Drive, Boulder Creek, CA 95006 (US). FANG, Emerson, S.; 1197 Wisteria Drive, Fremont, CA 94539 (US). GONG, Zhaoyu, G.; 1342 S. Stelling Road, Cupertino, CA 95014 (US). HO, Thomas, Y.; 40732 Ondina Place, Fremont, CA 94539 (US). HSU, Hengwei; 4209 Canfield Drive, Fremont, CA 94536 (US). LI, Sidong; 5598 LeFevre Drive, San Jose, CA 95118 (US).

NG, Sam; 34377 Maybird Circle, Fremont, CA 94555 (US). PAPAKIPOS, Matthew, N.; 1701 Oak Avenue, Menlo Park, CA 94025 (US). REDGRAVE, Jason, R.; 278 Martens Avenue, Mountain View, CA 95040 (US). TRIVEDI, Sushma, S.; 1208 Rembrandt Drive, Sunnyvale, CA 94087 (US). TUCK, Nathan, D.; 8666 Somerset Avenue, San Diego, CA 92123 (US).

(54) Title: GRAPHICS PROCESSOR WITH DEFERRED SHADING

(57) Abstract

A deferred graphics processor which includes deferred shading, a tiled frame buffer, multiple–stage hidden surface removal processing (Fig. 5), Phong shading (14000), subpixel anti–aliasing, and texture– and bump–mapping (12000).

GRAPHICS PROCESSOR WITH DEFERRED SHADING

Claim of Priority

This application claims the benefit under 35 USC § 119(e) of U.S. Provisional Application Serial No. 60/097,336 filed August 20, 1998, and under 35 USC § 120 of U.S. Patent Application Serial No. 09/213,990 filed December 17, 1998.

1.   Field of the Invention

This invention relates to computing systems generally, to three-dimensional computer graphics, more particularly, and more most particularly to structure and method for a three-dimensional graphics processor implementing differed shading and other enhanced features.

5

2.   Background of the Invention

2.1   Three-dimensional Computer Graphics

Computer graphics is the art and science of generating pictures with a computer. Generation of pictures, or images, is commonly called rendering. Generally, in three-dimensional (3D) computer graphics, geometry that represents surfaces (or volumes) of objects in a scene is translated into pixels stored in a frame buffer, and then displayed on a display device. Real-time display devices, such as CRTs used as computer monitors, refresh the display by continuously displaying the image over and over. This refresh usually occurs row-by-row, where each row is called a raster line or scan line. In this document, raster lines are numbered from bottom to top, but are displayed in order from top to bottom.

In a 3D animation, a sequence of images is displayed, giving the illusion of motion in three-dimensional space. Interactive 3D computer graphics allows a user to change his viewpoint or change the geometry in real-time, thereby requiring the rendering system to create new images on-the-fly in real-time.

- 2 -

In 3D computer graphics, each renderable object generally has its own local object coordinate system, and therefore needs to be translated (or transformed) from object coordinates to pixel display coordinates. Conceptually, this is a 4-step process: 1) translation (including scaling for size enlargement or shrink) from object coordinates to world coordinates, which is the coordinate system for the entire scene; 2) translation from world coordinates to eye coordinates, based on the viewing point of the scene; 3) translation from eye coordinates to perspective translated eye coordinates, where perspective scaling (farther objects appear smaller) has been performed; and 4) translation from perspective translated eye coordinates to pixel coordinates, also called screen coordinates. Screen coordinates are points in three-dimensional space, and can be in either screen-precision (i.e., pixels) or object-precision (high precision numbers, usually floating-point), as described later. These translation steps can be compressed into one or two steps by precomputing appropriate translation matrices before any translation occurs. Once the geometry is in screen coordinates, it is broken into a set of pixel color values (that is "rasterized") that are stored into the frame buffer. Many techniques are used for generating pixel color values, including Gouraud shading, Phong shading, and texture mapping.

A summary of the prior art rendering process can be found in: "Fundamentals of Three-dimensional Computer Graphics", by Watt, Chapter 5: The Rendering Process, pages 97 to 113, published by Addison-Wesley Publishing Company, Reading, Massachusetts, 1989, reprinted 1991, ISBN 0-201-15442-0 (hereinafter referred to as the Watt Reference).

Figure 1 shows a three-dimensional object, a tetrahedron, with its own coordinate axes $(x_{obj}, y_{obj}, z_{obj})$. The three-dimensional object is translated, scaled, and placed in the viewing point's coordinate system based on $(x_{eye}, y_{eye}, z_{eye})$. The object is projected onto the viewing plane, thereby correcting for perspective. At this point, the object appears to have become two-dimensional; however, the object's z-coordinates are preserved so they can be used later by hidden surface removal techniques. The object is finally translated to screen coordinates, based on $(x_{screen}, y_{screen}, z_{screen})$, where z_{screen} is going perpendicularly into the page. Points on the object now have their x and y coordinates described by pixel location (and fractions thereof) within the display screen and their z coordinates in a scaled version of distance from the viewing point.

Because many different portions of geometry can affect the same pixel, the geometry representing the surfaces closest to the scene viewing point must be determined. Thus, for each pixel, the visible surfaces within the volume subtended by the pixel's area determine the pixel color value, while hidden surfaces are prevented from affecting the pixel. Non-opaque surfaces closer to the viewing point than the closest opaque surface (or surfaces, if an edge of geometry crosses the pixel area) affect the pixel color value, while all other non-opaque surfaces are discarded. In this document, the term "occluded" is used to describe geometry which is hidden by other non-opaque geometry.

Many techniques have been developed to perform visible surface determination, and a survey of these techniques are incorporated herein by reference to: "Computer Graphics: Principles and Practice", by Foley, van Dam, Feiner, and Hughes, Chapter 15: Visible-Surface Determination, pages 649 to 720, 2nd edition published by Addison-Wesley Publishing Company,

Reading, Massachusetts, 1990, reprinted with corrections 1991, ISBN0-201-12110-7 (hereinafter referred to as the Foley Reference). In the Foley Reference, on page 650, the terms "image-precision" and "object-precision" are defined: "Image-precision algorithms are typically performed at the resolution of the display device, and determine the visibility at each pixel. Object-precision algorithms are performed at the precision with which each object is defined, and determine the visibility of each object."

As a rendering process proceeds, most prior art renderers must compute the color value of a given screen pixel multiple times because multiple surfaces intersect the volume subtended by the pixel. The average number of times a pixel needs to be rendered, for a particular scene, is called the depth complexity of the scene. Simple scenes have a depth complexity near unity, while complex scenes can have a depth complexity of ten or twenty. As scene models become more and more complicated, renderers will be required to process scenes of ever increasing depth complexity. Thus, for most renders, the depth complexity of a scene is a measure of the wasted processing. For example, for a scene with a depth complexity of ten, 90% of the computation is wasted on hidden pixels. This wasted computation is typical of hardware renderers that use the simple Z-buffer technique (discussed later herein), generally chosen because it is easily built in hardware. Methods more complicated than the Z Buffer technique have heretofore generally been too complex to build in a cost-effective manner. An important feature of the method and apparatus invention presented here is the avoidance of this wasted computation by eliminating hidden portions of geometry before they are rasterized, while still being simple enough to build in cost-effective hardware.

When a point on a surface (frequently a polygon vertex) is translated to screen coordinates, the point has three coordinates: 1) the x-coordinate in pixel units (generally including a fraction); 2) the y-coordinate in pixel units (generally including a fraction); and 3) the z-coordinate of the point in either eye coordinates, distance from the virtual screen, or some other coordinate system which preserves the relative distance of surfaces from the viewing point. In this document, positive z-coordinate values are used for the "look direction" from the viewing point, and smaller values indicate a position closer to the viewing point.

When a surface is approximated by a set of planar polygons, the vertices of each polygon are translated to screen coordinates. For points in or on the polygon (other than the vertices), the screen coordinates are interpolated from the coordinates of vertices, typically by the processes of edge walking and span interpolation. Thus, a z-coordinate value is generally included in each pixel value (along with the color value) as geometry is rendered.

2.2   Generic 3D Graphics Pipeline

Many hardware renderers have been developed, and an example is incorporated herein by reference: "Leo: A System for Cost Effective 3D Shaded Graphics", by Deering and Nelson, pages 101 to 108 of SIGGRAPH93 Proceedings, 1-6 August 1993, Computer Graphics Proceedings, Annual Conference Series, published by ACM SIGGRAPH, New York, 1993, Softcover ISBN 0-201-58889-7 and CD-ROM ISBN 0-201-56997-3 (hereinafter referred to as the